Transformation and simulation of domain specific languages[1]

Dávid Vágó

In the last decade, as computer applications have become more and more complex, modeling became an essential and unavoidable part of software development. Models represent software at a higher level of abstraction, allowing developers to focus mainly on the business intelligence, and delay implementation-specific questions to a later stage. In the early 1990s UML (Universal Modeling Language) was conceived to support the design of object-oriented systems. UML gives a language (graphical notation and semantics) to describe the structure and overall behavior of software components, but due to its imprecise semantics [1], formal verification of UML models is hardly possible. Its other weakness is that it tries to be as universal as possible, which makes application field specific modeling elaborate, generally done by extensions called *UML profiles*. Certain modeling aspects (such as reliability) already have a well-established formalisms (in case of reliability, that would be Petri nets). Thus a reliability experts, who had always worked with Petri nets, may find it difficult to transform their ideas to the language of UML.

Domain specific modeling (DSM) is a more advanced design methodology, which introduces the term *domain*. A domain is the set of concepts of a specific application field. For example, the domain of mobile communication includes the concepts *SMS*, *dialing*, *contact list*, etc. DSM allows the developers to create *domain-specific models*, models made up of these application-specific concepts. Thus an user interface designer would use familiar concepts like *window*, *scrollbar* or *menu* when designing the appearance of the system, and on the other hand, the reliability expert working on the same system could use Petri nets.

There are various DSM tools available (MetaEdit+, Microsoft DSL Tools and many others), and all of them provides support for graphical editing of domain-specific models and (generally template-based) source code generation. However there are two other important areas of domain-specific modeling, simulation and model transformations, which are not widely supported. Model-level simulation makes it easier for the developer to test the behavior of his/her model. Model transformation on the other hand is useful for formal model verification. Some tools make it possible to follow the execution of the generated code on the model level, but interactive simulation or model verification support is rare. In my paper, I focus on VIATRADSM, a tool designed by István Ráth and Dávid Vágó at the Department of Measurement and Information Systems of the Budapest University of Technology and Economics. This tool is based on VIATRA2[2], a universal model transformation framework made by other members of the same research group.

The underlying VIATRA2 framework has strong transformation capabilities (based on graph transformation rules and abstract state machines), and in my paper I examine how these capabilities may be used in a domain-specific modeling tool. My goal is to integrate interactive simulation and model transformation features into the existing VIATRADSM framework. Many existing DSM tools lack simulation support because they provide no way of describing model semantics. In the first part of my paper, I discuss how model semantics can be expressed using the declarative GTASM language of VIATRA2. Using GTASM, model semantics can be given using simple, pattern-based transformation steps. Using Petri nets as an example, I demonstrate how interactive simulation works in our DSM tool. Finally, I show that complex model transformations can be expressed and executed as simply as simulations, and I give an example how such a transformation can be used for model verification.

[1]This work was partially supported by the SENSORIA European project (IST-3-016004)

References

[1] D. Varró and A. Pataricza. VPM: A visual, precise and multilevel metamodeling framework for describing mathematical domains and UML (The Mathematics of Metamodeling is Metamodeling Mathematics), in: *Journal of Software and Systems Modeling*, October, 2003

[2] VIATRA2 Framework. *An Eclipse GMT Subproject* http://www.eclipse.org/gmt/.